

ASSEMBLING SASS

2

PROPERTY OF:
CODE SCHOOL

PROJECT VERSION:
PART TWO

STANDARD
INSPECTION : NO. 18



This product shall be subject to inspection by
Chief Engineer, Machine & Tools
and shall be approved by the
Executive Board. The project shall be
approved by the Board of Directors
and the Board of Directors shall be
responsible for the project. The project
shall be subject to inspection by the
Board of Directors.

S

| MORE SASS



Agenda

1. Sass Wrap-Up

2. Compass

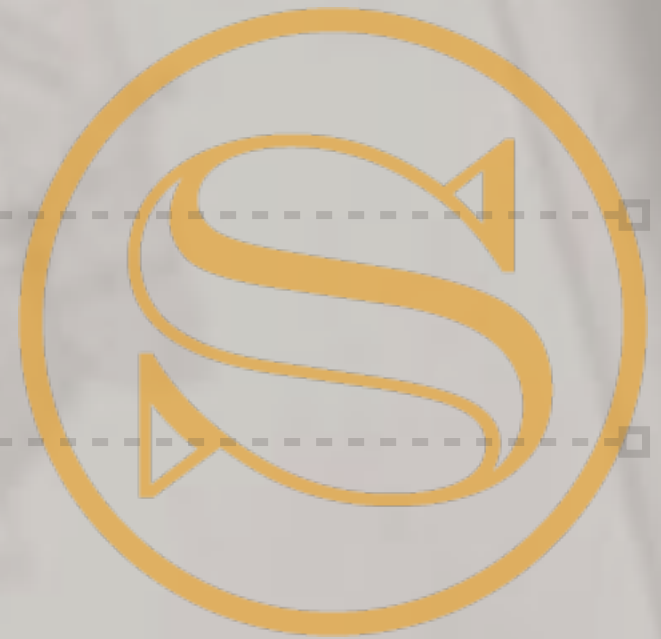


1.1 Indented Syntax

1.2 More Lists

1.3 More Utilities

1.4 More Placeholders



The Other Syntax

Indented syntax (.sass) relies on whitespace to simplify:

application.scss

```
.content {  
  border: 1px solid #ccc;  
  padding: 20px;  
  h2 {  
    font-size: 3em;  
    margin: 20px 0;  
  }  
}
```



application.sass

```
.content  
  border: 1px solid #ccc  
  padding: 20px  
  h2  
    font-size: 3em  
    margin: 20px 0
```

1.1 Indented Syntax



Indented Syntax + Mixins

Mixin declaration and use (optionally) become more efficient:

application.scss

```
@mixin box-sizing($x) {  
  -webkit-box-sizing: $x;  
  -moz-box-sizing: $x;  
  box-sizing: $x;  
}  
.content {  
  @include box-sizing(border-box);  
}
```



application.sass

```
=box-sizing($x)  
  -webkit-box-sizing: $x  
  -moz-box-sizing: $x  
  box-sizing: $x  
  
.content  
  +box-sizing(border-box)
```

1.1 Indented Syntax



Indented Syntax + This Course

- ⦿ Examples in this course will use the indented syntax
- ⦿ **Either** can be used on the challenges

1.1 Indented Syntax



Assembly Tip

The indented syntax **will**
feel strange at first, but give
it a fair shot*

* It took a while for us to warm up to it



List Refresher

A comma- **or** space-separated group of values:

application.sass

```
$authors: nick aimee dan drew  
$designers: jason, justin, tim
```

1.2 More Lists



application.sass

```
$authors: nick aimee dan drew  
  
@each $a in $authors  
  .author-#{ $a }  
    background: url(#{ $a }.jpg)
```

application.css

```
.author-nick {  
  background: url(nick.jpg)  
}  
.author-aimee {  
  background: url(aimee.jpg)  
}  
.author-dan {  
  background: url(dan.jpg)  
}  
.author-drew {  
  background: url(drew.jpg)  
}
```



1.2 More Lists



Length - number of items in a list:

```
$authors: nick aimee dan drew  
length($authors) // 4
```

Append - adds a value to the end of a list:

```
$authors: nick aimee dan drew  
append($authors, tim) // nick aimee dan drew tim
```

1.2 More Lists



Join - combines two lists together:

```
$authors1: nick aimee  
$authors2: dan drew  
join($authors1, $authors2) // nick aimee dan drew
```

Index - returns a value's list position or false:

```
$authors: nick aimee dan drew  
index($authors, dan) // 3  
index($authors, tim) // false
```

← starts at 1, not 0

1.2 More Lists



Nth - return the item at list position n:

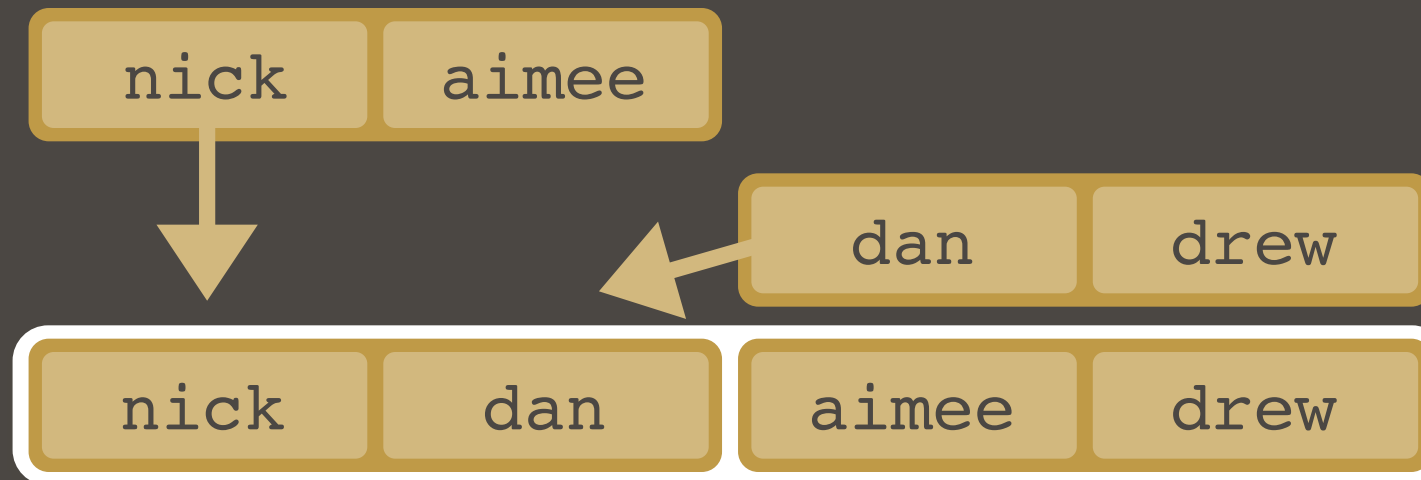
```
$authors: nick aimee dan drew  
nth($authors, 3) // dan
```

Zip - combines lists in comma-separated pairs:

```
$authors: nick aimee
```

```
$co-authors: dan drew
```

```
zip($authors, $co-authors)
```



1.2 More Lists



Using List Helpers

Pairing separate lists to create related pairs:

`application.sass`

```
$authors: nick aimee dan drew  
$colors: green blue red yellow  
  
$author-style: zip($authors, $colors)
```

nick

green

aimee

blue

dan

red

drew

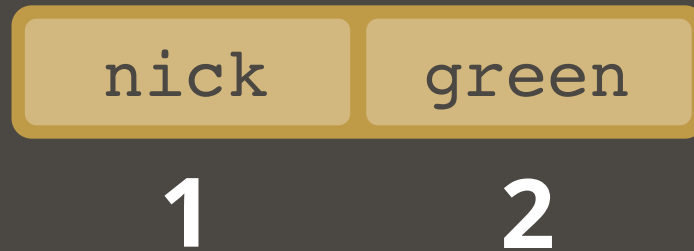
yellow

1.2 More Lists



application.sass

```
$authors: nick aimee dan drew  
$colors: green blue red yellow  
  
$author-style: zip($authors, $colors)  
  
@each $a in $author-style  
  .author-#{nth($a, 1)}  
    background: nth($a, 2)
```



application.css

```
.author-nick {  
  background: green;  
}  
.author-aimee {  
  background: blue;  
}  
.author-dan {  
  background: red;  
}  
.author-drew {  
  background: yellow;  
}
```

1.2 More Lists



List Length

Adding to the array requires a manual update to the `width`:

application.sass

```
$authors: nick aimee dan drew  
  
.author  
  float: left  
  width: 25%
```



application.sass

```
$authors: nick aimee dan drew tim  
  
.author  
  float: left  
  width: 25%
```

1.2 More Lists



Using `length()` keeps the `width` updated for us:

application.sass

```
$authors: nick aimee dan drew  
  
.author  
  float: left  
  width: 100% / length($authors)
```



application.css

```
.author {  
  float: left;  
  width: 25%;  
}
```

1.2 More Lists



1.1 Indented Syntax

1.2 More Lists

1.3 More Utilities

1.4 More Placeholders



Scaling Color

lighten and darken adjustments are linear:

application.sass

```
.content  
  color: lighten(#eee, 7%)
```



application.css

```
.content {  
  color: white;  
}
```

light colors reach
white at low percentages

1.3 More Utilities



Scaling Color

The `scale_color` function changes one or more color aspects **relative** to the start value:

- `$red`
- `$green`
- `$blue`
- `$saturation`
- `$lightness`
- `$alpha`

1.3 More Utilities



Scaling Color

Add keyword arguments for any aspects to be scaled:

application.sass

```
.content
  color: scale_color(#eee, $lightness: 7%)
```



application.css

```
.content {
  color: #efefef;
}
```

← 7% between #eee and white



1.3 More Utilities



Simple Conditionals

Part 1 included how to create an `if / else` chain in Sass:

application.sass

```
$theme: light

header
  @if $theme == dark
    background: #000
  @else
    background: #fff
```



application.css

```
header {
  background: white;
}
```

1.3 More Utilities



Single-line if - similar to a ternary operator:

```
if(condition, "true value", "false value")
```

1.3 More Utilities



application.sass

```
$theme: light

header
  @if $theme == dark
    background: #000
  @else
    background: #fff
```



application.sass

```
$theme: light

header
  background: if($theme == dark, #000, #fff)
```



1.3 More Utilities



Placeholder Selector Refresher

Can be extended; doesn't compile to a selector on its own:

application.sass

```
%btn
  border: 1px solid #ccc
  font-size: 1em
  text-transform: uppercase
```

1.4 More Placeholders



application.sass

```
%btn
  border: 1px solid #ccc
  font-size: 1em
  text-transform: uppercase

.btn-a
  @extend %btn
  background: #777

.btn-b
  @extend %btn
  background: #ff0
```

application.css

```
.btn-a, .btn-b {
  border: 1px solid #cccccc;
  font-size: 1em;
  text-transform: uppercase;
}
.btn-a {
  background: #777777;
}
.btn-b {
  background: yellow;
}
```



```
<a class="%btn" href="about.html">About</a>
```



1.4 More Placeholders



application.sass

```
%btn
.btn
  border: 1px solid #ccc
  font-size: 1em
  text-transform: uppercase

.btn-a
  @extend %btn
  background: #777

.btn-b
  @extend %btn
  background: #ff0
```

application.css

```
.btn-a, .btn-b,
.btn
  border: 1px solid #cccccc;
  font-size: 1em;
  text-transform: uppercase;
}
.btn-a {
  background: #777777;
}
.btn-b {
  background: yellow;
}
```



```
<a class="btn" href="about.html">About</a>
```

1.4 More Placeholders



application.sass

```
%btn
  border: 1px solid #ccc
  font-size: 1em
  text-transform: uppercase

.btn-a
  @extend %btn
  background: #777

.btn-b
  @extend %btn
  background: #ff0
```

index.html

```
<a href="#" class="btn-a">
  <b>Visit</b> Us
</a>
```



what if we need a nested
style on all %btn elements?

1.4 More Placeholders



application.sass

```
%btn
  border: 1px solid #ccc
  font-size: 1em
  text-transform: uppercase
  b
    color: #000

.btn-a
  @extend %btn
  background: #777
.btn-b
  @extend %btn
  background: #ff0
```

application.css

```
.btn-a, .btn-b {
  border: 1px solid #cccccc;
  font-size: 1em;
  text-transform: uppercase;
}
.btn-a b, .btn-b b {
  color: black;
}
.btn-a {
  background: #777777;
}
.btn-b {
  background: yellow;
}
```

1.4 More Placeholders

